Description

~~Method of Storing Data in a Random Access Memory, and an Encryption and Decryption Device~~METHOD OF STORING DATA IN A RANDOM ACCESS MEMORY AND ENCRYPTION AND DECRYPTION DEVICE

PRIORITY INFORMATION

This application claims priority from International application PCT/EP2004/012435, filed November 3, 2004 and German application 103 52 401.0, filed November 10, 2003.

BACKGROUND OF THE INVENTION

This invention relates in general to data security and in particular to ~~a method of~~ storing data in a random access memory~~, and to an encryption and decryption device~~.

~~In order t~~To ensure data security or to protect copyrights with respect to data stored in memory, a known approach is to store the data in encrypted form in a read-only memory (ROM), such as, for example, an EPROM, EEPROM, CD-ROM, or DVD-ROM~~, etc~~. These data may relate to both data from executable programs (program codes) as well as video or audio data. An approach is also known where~~by~~ video data or audio data are transmitted in encrypted form from a transmitting device to a receiving device.

~~——~~The ~~objective is that the~~ use of the encryption-stored or encryption-transmitted data is thereby theoretically enabled only for those users who have a corresponding decryption unit (decoder) with a "matching" key.

Conventional encryption algorithms, such as, for example, the DES method (~~DES =~~Data Encryption Standard) or the AES method (~~AES =~~Advanced Encryption Standard) encrypt/encode the data blockwise, wherein with the DES method, for example, 64 data bits are

encoded in one block. Since in the DES this-method the number of data bits contained in a data

block is usually greater thant the number of data bits of a data word capable of being processed

processable by a processing unit, it is necessary to have the processing unit first store the data

words obtained after decoding a data block in a random access memory (RAM) before these data

words undergo further processing.

These RAMs located externally to the processing unit represents a security risk insofar as

there is a possibility that the encrypted data can be tapped along the link between the RAM and

the processing unit. These data, for example video or audio data, can then be stored in

unencrypted form, thereby making them accessible to unauthorized use.

If the data stored in the RAM are happen to be the data of a program code, then there is

the risk that the program flow may be determined by unauthorized persons. In addition, there is

the risk that unauthorized program code may be fed into the unit executing the program in order,

for example, to provide additional functions which are not intended supposed to be provided by

the authorized program code.

What is needed is The goal of this invention is to provide a relatively secure technique

method of storing data in a RAM which does not have the aforementioned disadvantages and is

implementable at relatively low cost, as well as a device to encrypt and /decrypt the data stored

in a RAM.

SUMMARY OF THE INVENTION

These goals are achieved by a method according to Claim 1 and by a device according to Claim

12. Advantageous embodiments of the invention are described in the subordinate claims.

Briefly, In the method according to an aspect of the invention, a method for storing data

3

in a random access memory (RAM) in which data words are storable with a predetermined number of data bits, ~~involves~~ an encryption of each data word ~~is effected~~ before storage in the RAM, where~~by~~ a permutated data word with a predetermined number of data bits is generated from each data word, or from a data word derived therefrom, ~~this data word,~~ by a one-to-one rearrangement, or permutation of the individual data bits using a first permutation key.

~~An advantageous aspect of this method is that t~~The individual data bits of the permutated data word are substituted using a first substitution key before ~~the~~ storage, where~~in~~ the data word encrypted by permutation and subsequent substitution is stored in the RAM~~memory~~. ~~In this connection, t~~There is also the possibility of substituting the data bits of the data word to be encrypted before the permutation using a first substitution key, and of storing the data word obtained from the substitution and subsequent permutation as the encrypted data word.

The encryption of the individual data words is preferably performed ~~effected~~ in the same chip in which the processing unit that process~~es~~ing the data words is integrated. The data words transferred externally from this chip to the RAM ~~memory~~ for storage are provided in encrypted form ~~in this method~~, and are thus protected against interference effects or unauthorized tapping of the data. ~~In this method, t~~The encryption is performed ~~effected~~ data word by data word, with the result that, unlike the case of blockwise encryption, no additional storage on the chip is required for ~~the~~ encryption or ~~a~~ decryption.

The permutation or rearrangement of the individual data bits as determined by the permutation key represents an effective encryption method. Given a data word width of 32 bits, ~~width,~~ there are $32! \approx 2.6 \cdot 10^{35}$ different permutation possibilities. This number of permutation possibilities for a data word of 32 bit width~~length~~ increases by a factor of 2^{32} when in addition to

the permutation a substitution of the input data word, or of the already permutated data word, is performed ~~effected~~ using a substitution key of 32 bit width~~length~~.

The substitution of a data word ~~to be substituted~~ is performed ~~effected~~ as determined by the substitution key, for example, by assigning a key bit of the substitution key to each data bit of the data word, where~~in~~ the respective data bit is mapped, in unchanged or inverted form as a function of the value of the assigned substitution key bit, to the data word resulting from the substitution.

In one embodiment, the permutation key comprises a number of unique subkeys corresponding to the number of the data bits of the data word to be permutated, these keys each being assigned to a data bit of the data word resulting from the permutation. The individual subkeys indicate which of the data bits of the data word to be permutated is to be mapped to the respective data bit to which the subkey is assigned.

Each subkey of the permutation key ~~here~~ comprises a number of key bits, where~~in~~ preferably provision is made to implement incrementally the mapping of a data bit of the data word to be permutated to a data bit of the permutated data word using a subkey according to the following steps:

a) selecting a first group of data bits from the data bits of the permutated data word as determined by a first key bit of the subkey;

b) selecting a second group of data bits from the first group of data bits obtained by the previous selection as determined by a second key bit of the subkey; and

c) repeating step b), each time using an additional key bit ~~in order~~ to select from the group obtained by the previous selection an additional group until the selected group comprises only one more data bit which corresponds to the data bit of the permutated data word.

This type of incremental selection procedure to map a data bit of the data word to be permutated to a data bit of the permutated data word provides the advantage that no storage elements are required for ~~to~~ implementation ~~it~~.

The permutation key, and possibly the substitution key, are regenerated before a new writing to the RAM ~~memory~~, for example, after connection to a device containing the RAM ~~memory~~.

The substitution key, which comprises a number of substitution key bits, corresponding to the number of data bits, may be ~~is~~ generated ~~here~~ by picking out a corresponding number of bits from a sequence supplied by a random number generator.

When generating the permutation key, ~~care must be taken that~~ the individual subkeys preferably differ ~~so as~~ to ensure a one-to-one assignment of a data bit of the data word to be permutated to a data bit of the permutated data word. ~~In order t~~To generate the individual sub-permutation-keys which are each assigned to a bit position of the permutated data word, and which together yield the permutation key, provision is made to generate a sub-permutation-key consecutively for each bit position of the permutated data word, and thereby to check whether the generated sub-permutation-key has already been generated for another bit position. If this sub-permutation-key has already been generated, it is rejected and a new sub-permutation-key is randomly generated for the given bit position. If the randomly generated sub-permutation-key does not yet exist, then this key is retained for the given bit position. This procedure repeats until for ~~to~~ each bit position of the permutated data word one sub-permutation-key has been assigned for the selection of a data bit of the data word to be permutated.

The decryption of the data words stored in the RAM is effected analogously to the encryption procedure. If in a two-step procedure comprising permutation and substitution the

data word to be encrypted is first permutated and then substituted, then during decryption the encrypted data word is first "back"-substituted using a second substitution key to undo the substitution effected during encryption, and subsequently "back"-permutated using a second permutation key in order to undo the permutation effected during the encryption.

If during encryption of the data word first a substitution and then a permutation are performedeffected, then during decryption the encrypted data word is first permutated using the second permutation key, then substituted in order to recover the original data word.

Depending on the type of substitution used, the first substitution key can be selected in identical form to the second substitution key, for example, whenever the substitution comprises the consists in mapping of the individual data bits unchanged or inverted as determined by the key bits of the substitution key.

The following employs embodiments to explain the invention in more detail based on the figures.

These and other objects, features and advantages of the present invention will become more apparent in light of the following detailed description of preferred embodiments thereof, as illustrated in the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG.igure 1 is a block diagram illustration of shows an arrangement comprising an encryption and decryption unit arrangement which encrypts the data to be stored in a RAM random access memory and which decrypts the data read out from the RAMrandom access memory;.

FIG~~ure~~. 2 ~~shows an embodiment~~ is a block diagram illustration of ~~an~~ the encryption and decryption unit of FIG. 1; ~~arrangement comprising an encryption unit, a decryption unit, a key generator, and a random number generator.~~

FIG~~ure~~. 3 ~~shows an embodiment~~ is a block diagram illustration of ~~an~~ the encryption unit of FIG. 2; ~~arrangement which comprises a permutation unit and a substitution unit.~~

FIG~~ure~~. 4 ~~schematically illustrates the structure~~ is a block diagram illustration of ~~a~~ the permutation unit of FIG. 3; ~~which comprises selection units.~~

FIG~~ure~~. 5 ~~shows an embodiment~~ is a block diagram illustration of one of the ~~a~~ selection units of the permutation unit of FIG. 4; ~~which comprises multiple selection stages with selection switches.~~

FIG~~ure~~. 6 illustrates the functional principle of ~~a~~ the selection unit of FIG. 5 for a data word of 8 bit width~~;~~.

FIG~~ure~~. 7 ~~shows the circuit-logic-implemented embodiment~~ is a block diagram illustration of one of the selection switches of the selection unit of ~~shown in~~ FIG~~ure~~. 5~~;~~.

FIG~~ure~~. 8 ~~schematically illustrates an embodiment~~ is a block diagram illustration of the substitution unit of ~~shown in~~ FIG~~ure~~. 3~~;~~. ~~the substitution unit comprising multiple substitution elements.~~

FIG~~ure~~. 9 ~~illustrates a possible embodiment~~ is a block diagram illustration of one of the substitution elements of the substitution unit of FIG. 8; ~~shown in Figure 8.~~

FIG~~ure~~. 10 illustrates the construction of the permutation key from subkeys and key bits, and the construction of the substitution key~~;~~.

3000-0052

FIG.~~igure~~ 11 is a block diagram illustration ~~illustrates the complete structure~~ of a permutation unit of FIG. 2 for use in encrypting a ~~an encryption unit as indicated in Figure 2 for~~ data words of ~~4~~four bits~~;~~.

FIG.~~igure~~ 12 is a block diagram illustration of ~~shows the permutation unit corresponding to the~~a permutation unit of ~~shown in~~ FIG.~~igure~~ 2~~11~~ for use in decrypting a data word of four bits~~a decryption unit as indicated in Figure 2;~~ and

FIG.~~igure~~ 13 is a block diagram illustration ~~schematically illustrates the structure~~ of an internal memory~~,~~ provided in the key generator~~,~~ that ~~to~~ stores a first permutation key for the encryption of FIG. 11 and a second permutation key for the decryption of FIG. 12.

DETAILED DESCRIPTION OF THE INVENTION

Unless otherwise indicated, like ~~identical~~ reference numerals designate corresponding components and signals throughout the different views. ~~notations in the figures denote components and signals of identical meaning.~~

FIG.~~igure~~ 1 illustrates ~~shows~~ a random access memory (RAM) 20 which ~~is designed to~~ stores data words of n-bit ~~length~~ width. The RAM ~~Memory~~ 20 has an input 21 to read in data words to be stored, and an output 22 to read out stored data words. Not illustrated ~~shown~~ in FIG.~~igure~~ 1 are the well-known required control wires through which the memory addresses are communicated to the RAM 20~~memory~~, at which addresses the individual data words are to be stored or from which addresses the individual data words are to be read out.

Processing of the data words read into ~~memory 20,~~ or ~~read~~ out of the RAM 20 ~~this memory,~~ is performed ~~effected~~ in a data processing unit 30, for example, a processor. Depending on the type of the~~is~~ processor 30, the data words stored in the RAM ~~memory~~ 20 are, for example,

9

data words of a program code which is executed by the processor 30, or data words of video or audio data which are moved by the processor 30 through suitable output units for playback. in order to be perceived.

The Ddata processing unit 30 and the RAM memory 20 are not integrated on a common chip or integrated circuit ("IC"), as indicated in FIG.igure 1 by the broken line between the data processing unit 30 and the RAM memory 20. In order tTo prevent any "wiretapping" of or interference with data communication between the data processing unit 30 and the RAM memory 20, an encryption and decryption unit 10 is provided between the data processing unit 30 and the RAM memory 20 on the same chip on which the data processing unit 30 is located. The encryption/decryption unit This device 10 encrypts data words M outputted by the data processing unit 30 so as to provide encrypted data words M' which are stored word-by-word in the RAM memory 20. In the reverse direction, the encryption/decryption unit device 10 decrypts the data words M' stored in encrypted form in the RAM memory 20 in order to recreate the original data words M processed processable by the data processing unit 30. In FIG.igure 1 and subsequently, M denotes an arbitrary unencrypted data word of width length n, while M' denotes an arbitrary encrypted data word of width length n generated by encrypting a data word M.

FIG.igure 2 schematically illustrates the structure of the this encryption and decryption unit device 10 in more detail. The unit 10 device shown comprises an encryption unit 11 which has an input 110 of n-bit width to receive supply an unencrypted data word M, and an output 111 that provides to output an encrypted data word M'. Encryption of the data word M is performed effected as determined by a first key C which is provided by a key generator 13. To For the purpose of supplying this first key C, a binary random sequence RS is fed by a binary random number generator 12 to the key generator 13.

3000-0052

The encryption/decryption unit Device 10 further comprises a decryption an encryption unit 11' with an input 110' to supply an encrypted data word M' of n-bit width, and an output 111' to supply the decrypted data word M generated from the encrypted data word M'. The decryption is performed effected as determined by a second key C' which is matched to the first key C and which is also provided by the key generator 13.

The encryption decryption unit 11 maps the data word M using the first key C uniquely to the encrypted data word M', wherein:

$$M' = E(M,C) \qquad (1),$$

where E stands for the encryption function implemented by the encryption unit 11. Analogously:

$$M = D(M',C') \qquad (2),$$

where D stands for the decryption function implemented by the decryption unit 11'.

FIG.igure 3 schematically illustrates in more detail an embodiment of the encryption unit 11 of FIG. 2 which in the example comprises a permutation unit 14 and a substitution unit 15. The Ppermutation unit 14 has inputs to receive supply the individual data bits M[n-1]...M[0] of the data word M, and has outputs to supply data bits Mp[n-1], Mp[k], Mp[0] of a permutated data word Mp. The individual data bits Mp[n-1]...Mp[0] of the permutated data word Mp result from the data bits M[n-1]...M[0] of the data word M by permutation or /rearrangement as determined by a permutation key P. The permutation may be performed here is effected on a one-to-one basis, that is, one data bit each of the unencrypted data word M is mapped to one data bit of the permutated data word Mp.

In the example, the data bits Mp[n-1]...Mp[0] of the permutated data word Mp are then substituted by a substitution unit 15 as determined by a substitution key S, wherein the

11

substitution unit 15 provides the data bits of the encrypted data word M'. As determined by the substitution key S, one data bit each of the permutated data word Mp is mapped by the substitution unit 15 to one data bit M'[n-1]...M'[0] of the encrypted data word M'.

The following explains the structure and the functional principle of the permutation unit 14 with respect to ~~based on~~ FIGigures. 4-7. ~~5 through 7.~~ Also, ~~Next,~~ the structure and functional principle of the substitution unit 15 is ~~will be~~ explained with respect to FIGs. 8-9.~~based on Figures 8 and 9.~~

With reference to FIG.igure 4, the permutation unit 14 has a number of selection units 14_n-1...14_0 corresponding to the number of data bits of the data word M to be encrypted. ~~M,~~ ~~wherein a~~All of the data bits M[n-1]...M[0] of the data word M to be encrypted ~~M~~ are supplied to each of these selection units.~~, and wherein t~~The individual selection units 14_n-1...14_0 each provide a data bit Mp[n-1]...Mp[0] of the permutated data word Mp. Mapping of one of the data bits of the unencrypted data word M to one of the data bits of the permutated data word Mp is performed ~~effected~~ in the selection units 14_n-1...14_0 as determined by sub-permutation-keys P[n-1], P[k], P[0]. Each of these sub-permutation-keys differ ~~in order~~ to map each of the data bits of the input data word M exactly once to a data bit of the permutated data word Mp. The sub-permutation-keys together produce the permutation key P, where~~:~~

P = (P[n—1], ... P[0]).

The individual selection units 14_n-1...14_0 are structured identically, the structure of ~~a~~ ~~random~~ one of these selection units, for example, the ~~here~~ selection unit 14_k, ~~being~~ explained below with respect to ~~based on~~ FIG.igure 5.

————The~~is~~ selection unit 14_k <u>(FIG. 4)</u> provides the data bit Mp[k] from the data bits M[n-1]...M[0] of <u>the</u> data word M as determined by <u>the</u> sub-permutation-key P[k]~~,~~<u>,</u> <u>which</u> ~~This sub-permutation key~~ comprises m key bits P[k,m-1]...P[k,0].

<u>Referring to FIG. 5, the</u> ~~S~~selection unit 14_k comprises multiple selection stages 141_0...141_m-1. All <u>of</u> the data bits of <u>the</u> input data word M are supplied to a first selection stage 141_0. As determined by a first key bit P[k,0] of <u>the</u> sub-permutation-key P[k], th~~e~~<u>is</u> first selection stage 141_0 selects a first group of data bits which are supplied to a second selection stage 141_1. As determined by a second key bit P[k,1], <u>the</u> second selection stage 141_1 generates from this first group <u>of data bits</u> a second group <u>of data bits</u> which is supplied to the third selection unit 141_2.

In the example <u>illustrated in FIG. 5,</u> ~~shown,~~ reduction of the data bits present in the respective groups is <u>performed</u> ~~effected~~ from selection stage to selection stage by a factor of 2, such that after $m = \log_2(n)$ selection stages only one data bit is left which corresponds to data bit Mp[k] of <u>the</u> permutated data word Mp. In this example in which $n = 32 = 2^5$, there are thus $m = 5$ selection stages.

<u>Also, i</u>~~I~~n the example <u>of FIG. 5</u>, each of the selection stages comprises a number of selection switches 142, to which two data bits each of a data group are supplied, and which, as determined by a permutation key bit, select one of the two data bits and pass it on to the next selection stage.

————The supply of the individual data bits to the selection switches of <u>each of</u> the ~~given~~ selection stage<u>s</u> is <u>performed</u> ~~effected~~ such that two data bits each are supplied to a selection switch, which data bits have successive bit positions in relation to the group from which the ~~given~~ selection stage has made a selection. In the example of FIG.~~igure~~ 5, the respective higher-order bit is supplied to a first input IN1 <u>of the selection switch 142</u>, while the respective lower-

order bit is supplied to a second input IN2~~,~~ of the ~~given~~ selection switch 142. In the example shown, ~~for given~~ a key bit "1", the bit applied at the_input IN1~~, that is, the higher-order bit,~~ is passed to output OUT1~~,~~ and ~~thus~~ to the next selection stage.

The functional principle of the selection stage illustrated ~~shown~~ in FIG.~~igure~~ 5 is explained below based on an 8-bit-wide data word M with respect to ~~in~~ FIG.~~igure~~ 6. From these eight ~~8~~ data bits M[7]...M[0], one bit is selected to generate the_data bit Mp[k] of the permutated data word. The first key bit P[k,0] of the_subkey P[k] has a value of 1 so that out of two data bits that are consecutive in terms of significance the higher-order data bit ~~one~~ is selected, thus yielding a first group with data bits M[7], M[5], M[3], and_M[1]. Out of each two consecutive data bits, in terms of their significance~~,~~ ~~data bits, that is,~~ (i.e._data bits M[7], M[5] and M[3], M[1]), one data bit each is selected as determined by the second key bit P[k,1]. In the example, this key bit is "0", so that in each case the lower-order one of the two data bits is selected, that is, data bits M[5], M[1]. Out of this resulting additional group of data bits, one data bit is selected, in this case the higher-order ~~one or~~ data bit M[5], ~~is selected~~ as determined by the third key bit P[k,2] ~~in order~~ to generate the_data bit Mp[k] of the permutated data word.

If ~~one arranges~~ the data bits in each of the selection groups are arranged as a function of their significance, and out of two adjacent ones in terms of their significance given a key bit "1" ~~one selects~~ the higher-order data bit is selected, and given a key bit "0" ~~one selects~~ the lower-order one of these two data bits is selected, then the value of the bit position of the selected data bit, in this case of data bit M[5], corresponds to the decimal equivalent of the_subkey P[k], as explained below~~.~~

If the ~~one views~~ subkey P[k] is viewed as a binary numerical sequence, the most significant bit (MSB) of which is generated by the key bit P[k,m-1] of the last selection stage,

14

and the least significant bit (LSB) of which is generated by key bit P[k,0] of the first selection stage, then the decimal equivalent of this binary sequence, in this case $101_2 = 5_{10}$, corresponds to the bit position of the data bit M[5] selected from the data word M.

A circuit-logic implementation of one embodiment of one of the selection switches 142 is illustrated ~~shown~~ in FIG.~~igure~~ 7. ~~In order t~~To implement the described selection function, the selection switch 142 comprises two AND gates, AND1, AND2, the outputs of which are supplied to an OR gate, OR1, where~~in~~ the output of this OR gate forms the output OUT1 of the selection switch 142. One each of the inputs IN1, IN2 to supply the data bits is supplied to one of the AND gates, AND1, AND2. The other input of the AND gate AND1 is coupled to the third input IN3 to supply a key bit, where~~in~~ this key bit is supplied in inverted form through an inverter INV1 to the other input of the AND gate AND2. When a logical "1" is applied at the third input IN3, the data bit applied at the first input IN1 is passed through the first AND gate bit AND1 and the OR gate OR1 to the output OUT1. Given a logical "0" at the third input IN3, the data bit at the second input IN2 is accordingly passed through the second AND gate AND2 and the OR gate OR1 to the output OUT1.

With reference to FIG.~~igure~~ 8, the substitution unit 15 comprises a number of substitution elements 15_n-1...15_0 corresponding to the number of data bits.~~;~~ ~~o~~One data bit of the data word to be substituted is ~~being~~ supplied to each of the substitution elements; in the example of FIG.~~igure~~ 3, that of the permutated data word Mp. The substitution key S, on the basis of which the substitution is ~~effected~~ performed, comprises n key bits S[n-1]...S[0], where~~in~~ one of these key bits S[n-1]...S[0] is supplied to each of the substitution elements. The Ssubstitution elements 15_n-1...15_0 are designed, as determined by the respective substitution key bit S[n-1]...S[0], to

output in unchanged or inverted form the data bit Mp[n-1]...Mp[0] supplied to the respective substitution element 15_n-1...15_0.

A circuit-logic implementation of an embodiment of the~~is~~ substitution element <u>15 </u>is illustrated ~~shown~~ in F<u>IG.</u>~~igure~~ 9. The substitution element 15_k comprises ~~a~~ first and second AND gate<u>s</u> AND3, AND4, and an OR gate OR2 connected following <u>the </u>AND gates AND3, AND4~~.~~<u>,</u> ~~at t~~<u>T</u>he output of <u>the </u>~~which~~ OR gate <u>OR2 provides </u>the substituted data bit~~ is provided~~. The substituted data bit is supplied to the substitution element through a first input IN4, and this data bit is supplied in inverted form by a first inverter INV2 to <u>the </u>first AND gate AND3, and in unchanged form to <u>the </u>second AND gate AND4. The respective substitution key applied at a second input IN5 of the substitution element is supplied to <u>the </u>first <u>AND </u>gate AND3 in unchanged form, and to <u>the </u>second AND gate AND4 in inverted form by a second inverter INV3. This arrangement ensures that given a substitution key bit "1" the data bit applied at <u>the </u>first input IN4 is provided in inverted form, and given a substitution key bit "0" this data bit is provided in unchanged form at <u>the </u>output OUT2.

In the embodiment of F<u>IG.</u>~~igure~~ 3, the encrypted data word M' is generated from <u>the </u>unencrypted data word M by permutation and subsequent substitution of <u>the </u>data word Mp resulting from the permutation. It is ~~of course understood that it is~~ also possible first to substitute <u>the </u>data word M using <u>the </u>substitution key S<u>~~M~~</u>, and then to permutate the resulting substituted data word using <u>the </u>permutation key P ~~in order~~ to arrive at the encrypted data word M'.

The determining factor for the efficacy of an encryption system is the number of different possible keys. In the example described, <u>the </u>key C to encrypt <u>the </u>data word M is composed of <u>the </u>permutation key P and <u>the </u>substitution key S. <u>The P</u>~~p~~ermutation key P comprises a number of subkeys corresponding to the number of data bits, the <u>width</u>~~length~~ of the subkeys being defined

by m=log$_2$(n). With reference to FIG. 10, the permutation key P can be viewed as a vector with n subkeys P[n-1]...P[0], or as an n x m matrix of individual subkey bits P[n-1,m-1]...P[0,0]. For data words of width n=32, the permutation key P comprises 32 different subkeys P[n-1]...P[0], thereby resulting in 32! different key combinations. Given that for the substitution key S there are 2^n available possibilities, then for the number N possible keys C for data words to be encrypted of width n=32 the result is: N = (32!)·2^{32}.

The substitution key S for encryption and decryption can be generated as part of a binary random sequence.

A method of generating the permutation key P is explained below for a data word of width n=4 bit based on FIGs. 11-13.

FIG. 11 illustrates a first permutation unit 14 that generates the permutated data word Mp from the data word M with n=4 selection units 14_3, 14_2, 14_1, 14_0 which are each of two-stage form (m=log$_2$4=2).

FIG. 12 illustrates a second permutation unit 14' corresponding to the permutation unit 14 of FIG. 11 which functions to undo the permutation effected by the first permutation unit 14 as it decrypts the data word in the decryption unit 11 (FIG. 3). The second permutation unit 14' is identical to the first permutation unit 14 in structure and comprises four selection units 14'_3, 14'_2, 14'_1, and 14'_0. Each of these selection units 14'_3 ... 14'_0 functions to map one of the data bits Mp[3]...Mp[0] of the permutated data word Mp back to one of the data bits M[3]...M[0] of the original data word M. This selection of one of the data bits in the individual selection units 14'_3...14'_0 is performed in each case as determined by the subkeys P'[3]... P'[0] of a second permutation key P'. In the example

17

illustrated, ~~shown~~ P' =(P'[3], P'[2], P'[1], P'[0]), where the individual subkeys P'[3]...P'[0] each comprises~~ing~~ two subkey bits P'[3,1]...P'[0,0].

The generation of the subkeys P[3]... P[0] of the first permutation key P and of the associated subkeys P'[3]...P'[0] of the second permutation key P' is explained ~~below~~ based on FIG.~~igure~~ 13.

To generate the first and second permutation keys P, P', the key generator 13 (~~13 in~~ FIG.~~igure~~ 2) comprises a first and second key memory 131, 131', as well as an assignment register[+] 132. The ~~K~~key memories 131, 131' ~~are~~ each ~~designed to~~ store n subkeys of key width m=$\log_2(n)$. Given n=4, four subkeys of width~~length~~ 2 are storable in each of the key memories~~y~~ 131, 131'. Assignment of the subkeys stored in the first key memory 131 to the selection units 14_3...14_0, and thus to the individual data bits of the permutated data word Mp, is performed ~~effected~~ through the address of the key memory 131 which is addressable line-by-line and which in the example comprises n=4 lines. The memory address of a subkey in the~~is~~ first key memory 131 corresponds ~~here~~ to the bit position of the data bit of the permutated data word to which the respective key is assigned. A subkey P[k] at the memory address k of the key memory 131 is thus assigned to the kth data bit Mp[k] of the permutated data word Mp, where k represents one of the possible line addresses 0...n-1 of the memory.

Assignment of subkeys P'[3]...P'[0] of the second subkey P' to the selection units 14'_3 ... 14'_0 or to the data bits M[3]...M[0] of the original data word is performed ~~effected~~ analogously. That is, the subkey P'[k] stored at the memory position k of the second key memory 131' is assigned to the selection unit 14'_k and determines which of the data bits of the permutated data word Mp is to be mapped to the data bit M[k] at the kth position of the data word M.

Generation of the subkeys P[3]...P[0] of the first permutation key and of the second subkeys P'[3]...P'[0] is performed ~~effected~~ in a mutually matched fashion by a procedure ~~which is~~ explained below.

The subkeys of the first permutation key P are generated consecutively as random binary sequences of width ~~length~~ m=2 using the function generator 12 illustrated ~~shown~~ in FIG. ~~igure~~ 2. As explained, the individual subkeys ~~must~~ differ from one another ~~in order~~ to obtain a one-to-one assignment of the data bits of the data word M to be permutated ~~M~~ to the data bits of the permutated data word Mp. In the example described based on FIGs. ~~igures~~ 11 and 12, there are n=4 different subkeys which can be assigned randomly to the four selection units.

One memory position of the assignment register 132 is assigned to each of the possible different subkeys, in this case, "11", "10", "01", "00" ~~,~~ ~~wherein a~~ A predetermined value is entered in the assignment register 132 at the respective position if the assigned subkey has already been generated at a memory position of the memory 131, and thus for one of selection units 14_3...14_0, ~~so as~~ to avoid ~~again~~ generating the same key at a different memory address, and thus for another selection unit 14_3...14_0.

In the example, the assignment of a certain one of the possible subkeys to a memory address of the assignment register 132 is performed ~~effected~~ by directly mapping the value represented by the subkey to the address of the memory position of the ~~mapping memory~~ assignment register 132. For example, the memory position $10_2=2$ of the assignment register ~~memory~~ 132 is thus assigned to a subkey "10". If $P[k]=w_{n-1} ... w_0$ applies for a subkey, then for the address assigned to this subkey:

19

$$W = \sum_{i=0}^{i=n-1} w_i 2^i$$

~~In order t~~To generate the permutation key, the respective subkeys are randomly generated consecutively for the individual memory addresses of the first permutation key memory 131, wherein after generation of a given subkey a determination is made based on examination of the assignment register whether such a subkey has already been generated. If such a subkey has already been generated, the subkey is rejected and a new subkey is randomly generated. This procedure is repeated until subkeys have been generated for all the memory positions, and thus for all the selection units of the permutation unit 14.

When one of the possible subkeys is generated for the first time, a certain value, for example a "1," is entered at the memory address, assigned to this key, of the assignment register ~~memory~~ 132. If this subkey is randomly generated once again for another memory position of the memory 131, this is detected in the assignment register ~~memory~~ 132 based on the value entered, and the subkey is rejected for this different memory position.

As explained above, the binary value of a subkey P[3]...P[0] which is assigned to a selection unit 14_3...14_0 or to a data bit Mp[3]...Mp[0] of the permutated data word Mp corresponds to the data position of the data bit M[3]...M[0] of the input word M selected by the respective selection unit. Accordingly, the subkeys P'[n-1]...P'[0] of the second permutation key P' each indicate which of the data bits of the permutated data word Mp is to be mapped to the data bit M[3]...M[0] to which the respective subkey is assigned.

If the general condition applies that a subkey P[k] assigned to the kth data bit Mp[k] of <u>the</u> permutated data word Mp maps the ith data bit M[i] of the permutated data word to this data bit of <u>the</u> permutated data word Mp, then, conversely, the subkey P'[i] assigned to the ith data bit must map the kth data bit of <u>the</u> permutated data word Mp to this data bit.

<u>The</u> <s>S</s>second key memory 131' is organized analogously to <u>the</u> first key memory 131<s>.</s>; <s>t</s><u>T</u>hat is, the addresses at which the individual subkeys P'[n-1]...P'[0] are stored correspond to the bit positions of the data bits M[n-1]...M[0] to which the individual subkeys are assigned.

<s>In order t</s><u>T</u>o generate a matching subkey of <u>the</u> second permutation key P' for a randomly generated subkey P[k] of <u>the</u> first permutation key P, which subkey is assigned to the kth data bit of <u>the</u> permutated data word Mp, the address value k of <u>the</u> first subkey P[k] is entered at the address in <u>the</u> second key memory 131', the value of which corresponds to the binary value i represented by the first key<s>.</s>, <u>that is,</u> <s>In other words,</s> for P[k]=i, P'[i]=k.

Generation of the first and second permutation keys can be <u>performed</u><s>described</s> by the following <u>routine</u><s>algorithm</s>:

Line 1: FOR k = (n-1) DOWNTO 0

Line 2: Fetch random number from generator and compute i

Line 3: Check if MapReg (i) = 1, if true, go to Line 2

Line 4: Set MapReg(i) = 1

Line 5: Set o_store(k) = i

Line 6: Set i_store(i) = k

Line <u>7</u><s>4</s>: NEXT k.

MapReg(i) here represents the value at address k of the assignment register 132. The expression o_store(k) represents the value at address k of the first memory 131, while i_store(i) represents the value at address i of the second memory 131'.

As explained above, the permutation performedeffected during encryption and analogously during decryption is augmented by a substitution as determined by a substitution key. This substitution can be effected performed either before the permutation or after the permutation, the procedure being effected performed in the reverse order during the decryption. If during encryption the substitution is effected performed after the permutation, then during decryption the re-substitution is effected performed before the permutation. During the above-described substitution in which, as determined by the substitution key bits, the respective assigned data bit is passed on either inverted or unchanged, the same substitution key used during decryption is used during encryption.

Although the present invention has been illustrated and described with respect to several preferred embodiments thereof, various changes, omissions and additions to the form and detail thereof, may be made therein, without departing from the spirit and scope of the invention.

What is claimed is:

List of reference notations

———

AND1-AND4	AND-gate
C,C'	key
IN1-IN5	inputs
INV1, INV2	inverter
M	data word
M[n-1]... M[0]	data bits
M'[n-1]...M'[0]	data bits of an encrypted data word
Mp[n-1]...Mp[0]	data bits of a permutated data word
OR1, OR2	OR-gate
OUT1, OUT2	outputs
P	permutation key
P[n-1]...P[0]	subkey of a permutation key
S	substitution key tes[2]
10	encryption and decryption unit
11	encryption unit
11'	decryption unit
13	key generator
14	permutation unit
14_n-1...14_0	selection unit
15	substitution unit
15_n-1...15_0	substitution units
20	random access memory, RAM
20	random number generator
21	input of the RAM
22	output of the RAM
30	data processing unit
110	input of the encryption unit
110'	input of the decryption unit

———

[2] Translators note: this does not appear to belong.

111	output of the encryption unit
111'	output of the decryption unit
112	key input of the encryption unit
112'	key input of the decryption unit
131	first permutation key memory
131'	second permutation key memory
132	selection register
141_n-1... 141_0	selection stages
142	selection switch